

[Clique aqui para ler pelo PDF e melhorar a experiência](#)

Primeiros Comandos e Tipos de Dados

```
#include <stdio.h>

int main() {
    printf("Hello World\n");
    return 0;
}
```

Depois de aprender o que são algoritmos e pseudocódigos, agora vamos começar a ver como eles são transformados em programas reais na linguagem C. Nessa aula, vamos dar os primeiros passos para entender os conceitos de **variáveis**, **constantes** e **tipos de dados**.

O que é a Linguagem C?

A linguagem C é uma das mais antigas e influentes linguagens de programação. Desenvolvida nos anos 1970, ela ainda é amplamente usada hoje, especialmente para o desenvolvimento de sistemas operacionais, aplicativos de alta performance e softwares embarcados. Por ser uma linguagem de baixo nível, C oferece ao programador mais controle sobre o hardware, o que a torna poderosa e eficiente.

Estrutura Básica de um Programa em C

Todo programa em C segue uma estrutura padrão que pode ser dividida em três partes: entrada, processamento e saída.

1. Entrada: Aqui, coletamos os dados que o programa precisa para funcionar.
2. Processamento: Onde as operações são realizadas (como cálculos ou tomadas de decisão).
3. Saída: Exibimos o resultado para o usuário.

Exemplo da estrutura básica do C

```
#include <stdio.h>

int main() {
    printf("Hello, World!\n");

    return 0;
}
```

Agora vamos detalhar o que cada parte no código acima faz:

1. `#include <stdio.h>`: Essa linha é responsável por incluir uma biblioteca padrão chamada `stdio.h`, que contém funções úteis para entrada e saída, como `printf`. Sempre que precisarmos usar funções como essas, precisamos incluir essa biblioteca.
2. `int main()`: O ponto de entrada de todo programa C é a função `main`. É aqui que o programa começa a executar. A função retorna um valor inteiro, que geralmente é 0 para indicar que o programa terminou com sucesso. É na `main()` onde chamamos nossas funções, onde são declaradas nossas variáveis, nossos condicionais, etc.
3. `printf("Hello, World!\n");`: Esta linha usa a função `printf` para imprimir texto na tela. O `\n` no final é um "caractere especial" que indica uma nova linha, ou seja, após imprimir "Hello, World!", o cursor será movido para a linha seguinte.
4. `return 0;`: Esta linha indica que o programa terminou com sucesso. O número 0 é retornado ao sistema operacional para informar que o programa executou corretamente. Retornos de funções serão vistos mais a fundo em aulas seguintes.

Conceito de Variáveis e Constantes

O que são Variáveis?

Variáveis são como "caixinhas" onde podemos armazenar dados temporariamente enquanto o programa está em execução. Imagine que você está fazendo uma conta e precisa guardar o valor de uma operação para usá-lo depois — as variáveis fazem exatamente isso. Elas podem armazenar diferentes tipos de dados, como números, letras e até valores verdadeiros ou falsos.

Exemplo simples em C:

```
int idade = 25; // Aqui, criamos uma variável chamada 'idade' que armazena o valor 25
```

O que são constantes?

Constantes são como variáveis, mas com uma diferença importante: o valor de uma constante não pode ser alterado depois que é definido. Elas são úteis quando temos valores que nunca mudam durante a execução do programa, como o valor de π . Como é de se imaginar, uma variável que seja utilizada para referenciar o valor de π não pode ter seu valor alterado visto que isso poderia resultar em erros grotescos de cálculos etc.

Exemplo em C:

```
const float PI = 3.14159 // Aqui, criamos uma variável chamada 'idade' que armazena o valor 25, note o termo const antecedendo a declaração
```

Tipos de Dados Primitivos

Na linguagem C, existem quatro tipos de dados principais que você deve conhecer:

1. Inteiros (int): Armazenam números inteiros, ou seja, sem decimais.

```
int numero = 42; // Exemplo de variável do tipo inteiro
```

2. Reais (float e double): Armazenam números com casas decimais.

```
float preco = 19.99; // Exemplo de variável do tipo float
```

3. Caracteres (char): Armazenam caracteres individuais, como letras e símbolos.

```
char letra = 'A'; // Exemplo de variável que armazena o caractere 'A', perceba que como a variável é do tipo textual ela possui sua atribuição entre ''
```

4. Booleanos (int): Armazenam valores lógicos (verdadeiro ou falso). Em C, usamos inteiros para representar booleanos: 0 é falso e 1 é verdadeiro.

```
int verdadeiro = 1; // Representa o valor 'true'  
int falso = 0; // Representa o valor 'false'
```

Exemplo de código abaixo unindo os tipos primitivos:

```
#include <stdio.h>  
  
int main(){  
    int idade = 20;  
    float peso = 88.5;  
    char letra = 'E';  
    int sobrepeso = 0;  
}
```

Operadores Aritméticos, Relacionais e Lógicos

Operadores Aritméticos

Estes operadores são usados para realizar operações matemáticas simples:

- + (soma)
- - (subtração)
- * (multiplicação)
- / (divisão)
- % (módulo ou resto da divisão)

Exemplo:

```
int soma = 10 + 5; // Resultado será 15
int subtracao = 10 - 5; // Resultado será 5
int multiplicacao = 10 * 5; // Resultado será 50
int divisao = 10 / 5; // Resultado será 2
int modulo = 10 % 5; // Resultado será 0 visto que 10 é divisível por 5 e possui resto 0
```

Observação: as operações divisão e módulo podem atribuir diferentes resultados a variável dependendo do seu tipo, mas isso será melhor explicado posteriormente.

Conclusão

Parabéns! Agora você já sabe o básico sobre variáveis, constantes, tipos de dados e operadores em C, além de entender como estruturar um programa simples. Pronto para o próximo passo? No próximo tutorial, vamos aprender sobre controle de fluxo, ou seja, como tomar decisões no seu programa usando estruturas condicionais.

[\[Voltar ao Topo\]](#)